Programmer's Learning Machine

Matthieu Nicolas

IJD Seminar, 2016-02-02

Outline

1. Presentation of PLM
   - Purposes
   - Demo
   - About PLM
   - Architecture
2. User's code's assessment
   - Challenges
   - Extraction of the execution component
   - Docker
3. Result
4. Next steps

Outline

1. Presentation of PLM
   - Purposes
   - Demo
   - About PLM
   - Architecture
2. User's code's assessment
   - Challenges
   - Extraction of the execution component
   - Docker
3. Result
4. Next steps

Presentation of PLM

Purposes

- Application to learn programming.

Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...

Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.

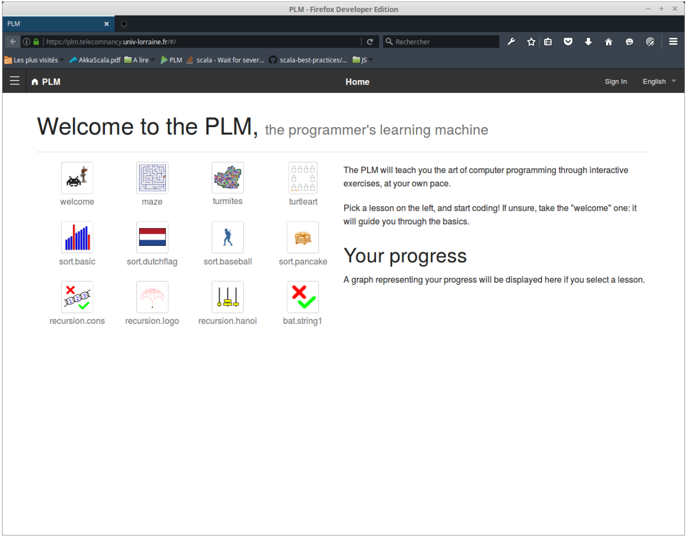
Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.
- Used at TELECOM Nancy since 2008.

Presentation of PLM

Quick demo



Welcome to the PLM, the programmer's learning machine

The PLM will teach you the art of computer programming through interactive exercises, at your own pace.

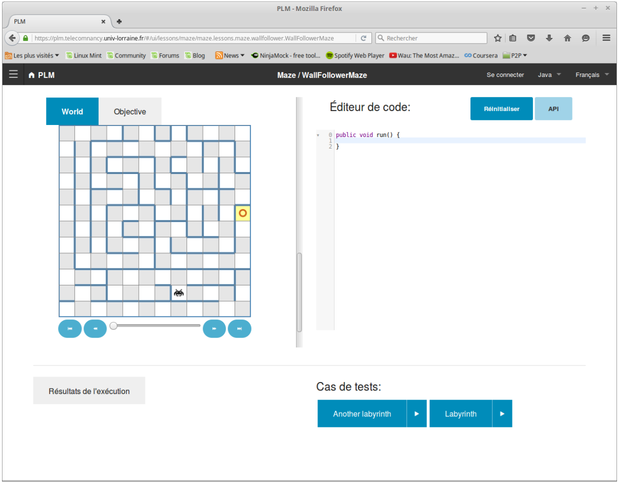
Pick a lesson on the left, and start coding! If unsure, take the "welcome" one: it will guide you through the basics.

Your progress

A graph representing your progress will be displayed here if you select a lesson.

Presentation of PLM

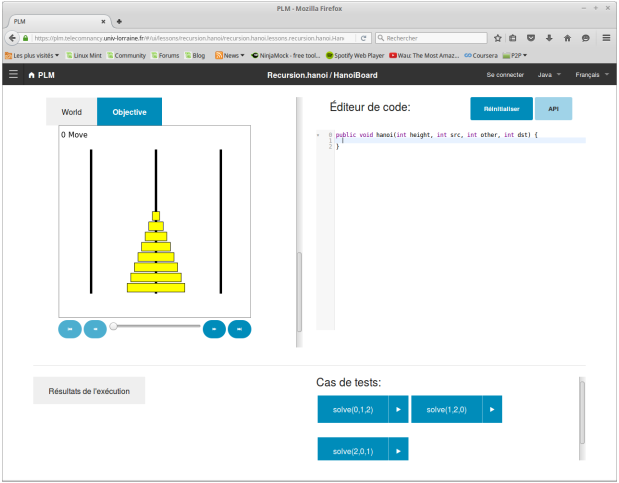
12 lessons, 200 exercises



```
public void run() {
}
```

Presentation of PLM

12 lessons, 200 exercises

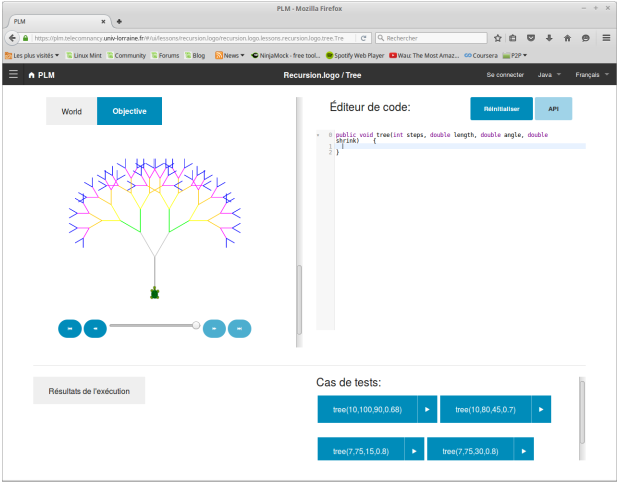


```
public void hanoi(int height, int src, int other, int dst) {

}
```

Presentation of PLM

12 lessons, 200 exercises



Recursion.logo / Tree

Éditeur de code:

```
public void tree(int steps, double length, double angle, double
shrink)    {
  
}
```

Cas de tests:

tree(10,100,90,0.68) ▸ tree(10,80,45,0.7) ▸

tree(7,75,15,0.8) ▸ tree(7,75,30,0.8) ▸

Presentation of PLM

Supported languages

- English
- French
- Brazilian Portuguese

Presentation of PLM

Supported programming languages

Presentation of PLM

Evolution of the project

- Formerly a fat client
    - Written in Java

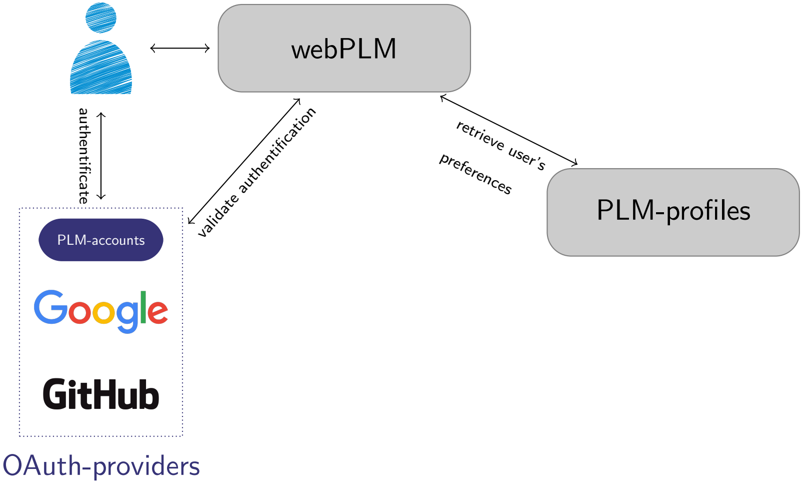
Presentation of PLM

Evolution of the project

- Formerly a fat client
  - Written in Java
- Switch to a web application
  - Server implemented in Scala using PlayFramework
  - User interface written in Javascript using AngularJS and Foundation

Presentation of PLM

Application's architecture

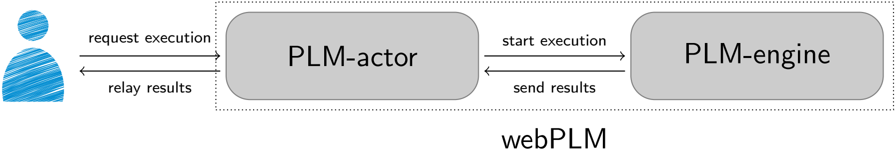


Outline

1. Presentation of PLM
    - Purposes
    - Demo
    - About PLM
    - Architecture
2. User's code's assessment
    - Challenges
    - Extraction of the execution component
    - Docker
3. Result
4. Next steps

User's code's assessment

Execution components



User's code's assessment

Limits

- Run on the same machine, same JVM

User's code's assessment

Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
    - Infinite loops

User's code's assessment

Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files

User's code's assessment

Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files
- And from *System.exit(whatever)*?

User's code's assessment

Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files
- And from *System.exit(whatever)*?
- Also need to scale.

User's code's assessment

Judges

- Delegate the execution to workers
- *Let it crash* strategy
- Allow to run code without impacting webPLM's performances
- Able to scale

User's code's assessment

Architecture with judges

- Need to:
    - deploy them easily
    - reset them to their original state
    - restrict their resources usage

User's code's assessment

Docker

User’s code’s assessment

Example of Dockerfile

User's code's assessment

Docker-compose

Outline

1. Presentation of PLM
    - Purposes
    - Demo
    - About PLM
    - Architecture
2. User's code's assessment
    - Challenges
    - Extraction of the execution component
    - Docker
3. Result
4. Next steps

Result

Current architecture

Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.

Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.
- Engine is (almost) working fine...

Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.
- Engine is (almost) working fine...
- ... but user experience needs to be improved!

Result

Live-session in TELECOM Nancy

- Can’t cope with the workload.

Result

Live-session in TELECOM Nancy

- Can’t cope with the workload.
- No tools for monitoring set up...

Result

Live-session in TELECOM Nancy

- Can't cope with the workload.
- No tools for monitoring set up...
- ... so the bottleneck is unknown.

Outline

1. Presentation of PLM
    - Purposes
    - Demo
    - About PLM
    - Architecture

2. User's code's assessment
    - Challenges
    - Extraction of the execution component
    - Docker

3. Result

4. Next steps

Next steps

Refactor the code

- Rushed to release a stable version before September...
- Needed to refactor some parts of the code.
- Standardized behavior of local and server mode.

Next steps

Simplify workflow to adapt the content

- Store most content inside PLM.
- Heavy and error prone workflow.
- Need to extract the content from PLM's jar.
- Allow to implement an exercise editor.

Next steps

Solve performance issues

- Set up some monitoring tools.
- Perform some load testing to identify the bottleneck.

Questions

Thanks for your attention, any questions?